

МЕДИЦИНАЛЫҚ ҚҰРАЛ-ЖАБДЫҚТАРДЫ ЖИНАУ ЖӘНЕ ЖӨНДЕУ ҮШІН ТОЛЫҚТЫРЫЛҒАН НАҚТЫЛЫҚ ҚОСЫМШАСЫН ДАЙЫНДАУ

Ниязова Р.С., Тлеубай Ж.

(Л.Н.Гумилев атындағы Еуразия ұлттық университетінің 2 курс магистранты)

Мақалада толықтырылған нақтылық технологиясын медицина саласында қолданудың қажеттілігі, сонымен қатар артықшылығы мен тақырып аясындағы қолдану тәсілі сипатталады.

Соңғы 15 жыл ішінде біздің жұмысымызда, қарым-қатынасымызда, көпшілік орындарда, күнделікті тұрмысызға деген көзқарасымыз және күнделікті өмірімізде айтарлықтай көп өзгерістер болды. Ал технологиялардың дамуы әлі аяқталған жоқ. Google, Facebook, әлемдегі сансыз мобильді қосымшалар мен бұлтты платформалар ұсынған бірдей технология қазір адамдардың өмірін жақсартуға және денсаулық емдеуге арналған. Соңғы бірнеше жылда денсаулық сақтау саласында үлкен жаңалықтар орын алса да, 2016-шы жыл медициналық IT индустриясы үшін ең үлкен жаңалықтардың бірі болды. Олардың ішінде денсаулық сақтау жүйелерінің үйлесімділігін қамтамасыз ететін қосымшалар, пациенттерді қашықтықтан бақылау, медбике көмекшісі бола алатын роботтар, қысым мен жараның алдын алатын электронды іш киімдер бар. Алайда, шын мәнінде денсаулық сақтау мен медицина салаларын қайта құрудың ең маңызды технологиясы ретінде толықтырылған нақтылық технологиясы баршама ғалымдар мен көпшіліктен қолдау тапқан болатын. Толықтырылған нақтылық технологиясы – виртуалды, яғни цифрлы техника саласында және нақты әлемнің ажырамас байланысын ұсынады, бұл оны денсаулық сақтаудың әртүрлі салаларында, құрал - жабдықтар мен емдеу процедураларында қолдануға өте ыңғайлы құрал етеді.

Бүгінгі таңда медициналық жабдықтардың көпшілігі өте күрделі, көп функционалды және алдыңғы қатарлы электроника мен бағдарламалық жасақтамаға ие, бұл оны қолдануда және жөндеу мен сүйемелдеу барысында күрделірек етеді. Осылайша, мұндай медициналық жабдықтың кез-келген операциялық немесе функционалды ақаулығы күрделі, сонымен қатар ақауларды табу мен жөндеу көп мөлшердегі қаражатты қажет етуі мүмкін. Әдеттегі өндірістік ақаулар жағдайы, құралдардың жұмысын тоқтата тұру - қаржылық немесе өндірістік шығындарды білдіреді, ал денсаулық сақтау саласында бұл тіпті қымбат, себебі адамдардың өмірін жоғалтуды білдіруі мүмкін. Осындай жағдайларда толықтырылған нақтылық қосымшасын пайдалану сияқты технологиялар, жабдықтың істен шығуын қарапайым етеді және әрдайым бақылауда ұстауға көмектеседі, бұл - тез жөндеуге, шығындарды азайтуға және тиімділікті арттыруға мүмкіндік береді. Сондықтан, толықтырылған нақтылық технологиясын игермеу, медицина саласындағы мәселелерді, яғни ақауларды жою процесін одан әрі ұзарып кетуін, клиникаға немесе аурухана ұжымына қолайсыздық тудыру сияқты мәселелерді ескермеу болып табылады.

Толықтырылған нақтылық технологиясы – денсаулық сақтау саласындағы құралдар жұмысын әр түрлі жолдармен бақылап және мінсіз жүргізуді кепіл етеді. Құралдарға қызмет көрсету кезінде, сервис техниктері қарапайым мәтінді немесе суретті интерактивті немесе аудиовизуальды ақпараттарға айналдыратын ұялы телефон немесе ақылды көзілдірік арқылы жөндей алады, себебі аталған технологиялар арқылы олар күрделі бөлшектердің нақты атаулары мен орналасу реті, орны сияқты ақпаратқа ие бола алады.

Ал медициналық жабдықтаманы қалпына келтіру процесінде толықтырылған нақтылық технологиясын қолдана отырып техникадағы ақаулардың сипаты беріліп, жөндеу нұсқаулығын әзірлеуге болады, бұл оны идеалды оқулыққа айналдырады. Сонымен қатар, дала техниктері кеңсенің мамандарымен қашықтан байланыс орната алады, бұл машинаның ағынында жұмыс істейді. Екеуі де сайттағы бейне ағындарын қолдана отырып бірлесіп жұмыс істей алады, бөлісілген бейнелер бойынша аннотация жасай алады және бір уақытта аудио қоңырауды қолдана отырып деректер алмасады, және ақауларды жою және жөндеу жұмыстарын бір уақытта аяқтай алады. Толықтырылған нақтылық қосымшалары монтаждау және техникалық қызмет көрсету уақытын 30% -ға қысқартады, орнату және жөндеу кезінде 20% қателіктерді болдырмайды және алыстан келетін мамандар үшін алыс сапарларды 70% қысқарады, бұл медицина саласындағы адам шығындарын азайтуға көмектеседі.

Денсаулық сақтау сияқты маңызды сегмент үшін толықтырылған нақтылық көптеген артықшылықтары оны медициналық орталықтар мен медициналық технологиялар саласындағы мамандар үшін қажет құрал етеді. Ауруханалар ішіндегі күрделі жүйелер мен механизмдерге қызмет көрсетуге кететін уақытты, шығындарды және ресурстарды азайтады. Сонымен қатар, технологиялық компаниялар нақты жабдықтар мен машиналар үшін ресурстар бөлудің орнына, әртүрлі жабдықты өңдей алатын техникалық қызмет көрсетушілердің әлдеқайда аз тобын қажет етеді.

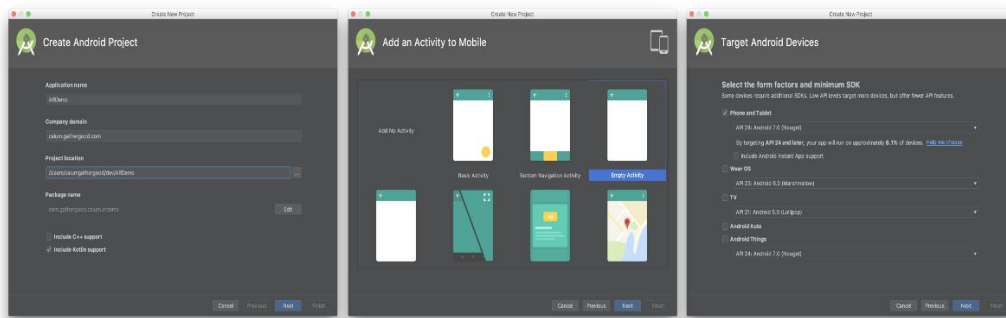
Мақала аты сәйкесінше жүргізіліп отырған диссертациялық жұмыс аясында жазылып отыр. Жоғарыда аталып өткен медицина саласындағы мәселелер мен оларды шешу жолдары мен аталған технологияны қолданудағы тиімділікті ескере отырып, жоба аясында Unity3D бағдарламасы арқылы жасалған медициналық құрылғы мысалы қорытындылайды. Аталған 3D модельды құрып мобильді қосымша ішінде көру, медициналық құрылғының барлық сипаттамасын көруге, тез орнатуға және сүйемелдеу мен қалпына келтіруге мүмкіндік береді.



1 сурет - Медициналық құрылғылардың 3D моделіне мысалдар

Қосымшада суретте мысал ретінде көрсетілгендей жұмысшылар ұялы телефондары арқылы кез келген сипаттасын қарап орнатып, сүйемелдеп, жөндей алады.

Мақала жазылып отырған диссертациясының жұмыс нәтижесі ретінде жоғарыда пайымдалған материалды мобильді қосымша түрінде дайындау. Мобильді қосымша Android Studio бағдарламалау ортасында java бағдарламалау тілінде sceneform технологиясы арқылы жасалады.



2 сурет - Android Studio басты парақшалары

Суретте көрсетілгендей android бағдарламасын ашып, келесідей бос жобаны таңдау қажет.

Біздің app /build.gradle файлында Sceneform-қа тәуелділікті қосуға болады - жобаны жаңарту үшін «Қазір синхрондау» түймесін басуды ұмытпаңыз. Біз бұл оқу құралын Котлинде жазатын болсақ та, Sceneform Java 8-тен кейбір тілдік конструкцияларды қолданады, сондықтан біз нақты қолдауды қосуымыз керек, себебі біздің ең төменгі API 26-дан аз.

```
android {
    ...
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    dependencies {
        ...
        implementation "com.google.ar.sceneform.ux:sceneform-ux:1.4.0"
    }
}
```

Бұл толықтырылған нақтылық қосымшасы болғандықтан, қолданушыдан камераға кіруді сұрауымыз керек. Сондай-ақ бізге AR рұқсаты қажет, яғни бұл бағдарлама ARCore-ны қажет етеді. Ақырында, ARCore қосылған құрылғысы жоқ пайдаланушыларға арналған бағдарламаларды сүзу үшін Play Store қолданатын арнайы AR мета-тег бар.

Келесі кодты Android Manifest файлына енгізуіміз қажет.

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.gathergood.calum.ardemo">
```

```
<uses-permission
android:name="android.permission.CAMERA" />
<uses-feature
android:name="android.hardware.camera.ar"
android:required="true" />
```

```

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl="true"
android:theme="@style/AppTheme">

<meta-data
android:name="com.google.ar.core"
android:value="required" />

<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

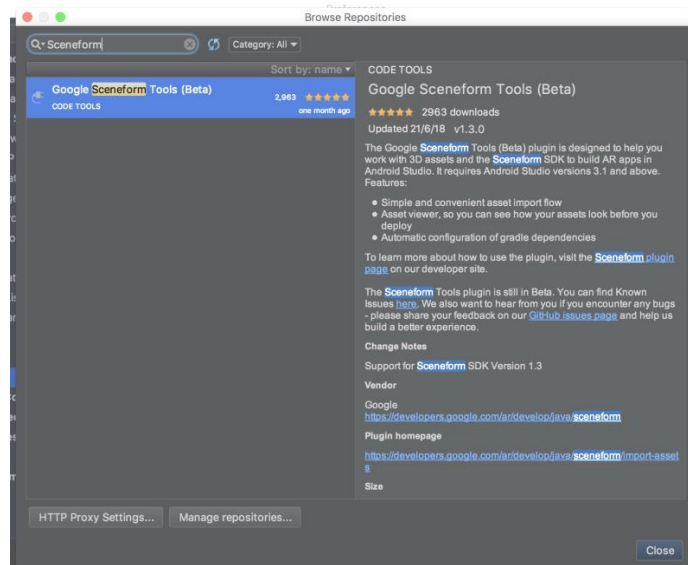
<category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>

```

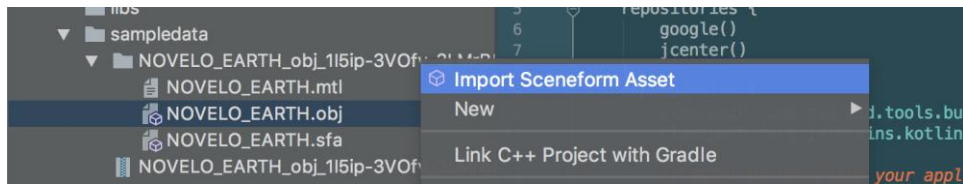
Үлгіні қосу

Біздің өмірімізді жеңілдету және өзіміздің 3D ресурстарымызды импорттауға мүмкіндік беру үшін Android Studio үшін Sceneform плагинін қосамыз. Мұны параметрлерді, плагиндерді таңдау, репозитарийлерді қарау және Google Sceneform құралдарын іздеу арқылы орнатуға болады.



3 сурет - Sceneform плагинін орнату мысалы

Келесі суретте қосымшаға 3D моделімізді қосамыз.



4 сурет - 3D модельге сілтеме беру

Бұл процесстен кейін, объектілерге сілтемені бағдарлама деңгейінде және активтер каталогында .sfb файлында таба аласыз. Бұл Sceneform екілік активі, біз оны қолданамыз. Sceneform Android Studio плагиінің нақты уақыт режимінде жаңадан құрылған активтің көріністерін көруге мүмкіндік береді.

Келесі қадам - Sceneform фрагментін макетімізге қосу. Сіз жай мәтіндік көріністі осы элементпен алмастыра аласыз. Біз сондай-ақ нақты сөзге моделімізді қосу үшін өзгермелі әрекет батырмасын (FAB) қосамыз.

```
<? xml version = " 1.0 " encoding = " utf-8 " ?>
< android.support.constraint.ConstraintLayout xmlns : android = "
http://schemas.android.com/apk/res/android "
xmlns : app = " http://schemas.android.com/apk/res-auto "
xmlns : tools = " http://schemas.android.com/tools "
    android : layout_width = " match_parent "
    android : layout_height = " match_parent "
    tools : context = " .MainActivity " >
```

```
< фрагмент
    android : id = " @ + id / sceneform_fragment "
    android : name = " com.google.ar.sceneform.ux.ArFragment "
    android : layout_width = " match_parent "
    android : layout_height = " match_parent "
    app : layout_constraintBottom_toBottomOf = " parent "
    app : layout_constraintEnd_toEndOf = " parent "
    app : layout_constraintStart_toStartOf = " parent "
    app : layout_constraintTop_toTopOf = " parent " />
```

```
< android.support.design.widget.FloatingActionButton
    android : id = " @ + id / floatingActionButton "
    android : layout_width = " wrap_content "
    android : layout_height = " wrap_content "
    android : layout_marginBottom = " 8dp "
    android : layout_marginEnd = " 8dp "
    android : layout_marginStart = " 8dp "
    app : layout_constraintBottom_toBottomOf = " @ + id / sceneform_fragment "
    app : layout_constraintEnd_toEndOf = " parent "
    app : layout_constraintStart_toStartOf = " parent "
    app : srcCompat = " @ drawable / ic_explore_white_24dp " />
```

```
</ android.support.constraint.ConstraintLayout >
```

Қосымшаны ақтау үшін келесідей java кодын негізгі активитиға қосамыз.

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var arFragment: ArFragment  
  
    private var isTracking: Boolean = false  
    private var isHitting: Boolean = false  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        arFragment = sceneform_fragment as ArFragment  
  
        arFragment.arSceneView.scene.addOnUpdateListener { frameTime ->  
            arFragment.onUpdate(frameTime)  
            onUpdate()  
        }  
  
        floatingActionButton.setOnClickListener {  
            addObject(Uri.parse("NOVELO_EARTH.sfb")) }  
            showFab(false)  
  
        }  
  
        private fun showFab(enabled: Boolean) {  
            if (enabled) {  
                floatingActionButton.isEnabled = true  
                floatingActionButton.visibility = View.VISIBLE  
            } else {  
                floatingActionButton.isEnabled = false  
                floatingActionButton.visibility = View.GONE  
            }  
        }  
  
        private fun onUpdate() {  
            updateTracking()  
  
            if (isTracking) {  
                val hitTestChanged = updateHitTest()  
                if (hitTestChanged) {  
                    showFab(isHitting)  
                }  
            }  
        }  
    }  
}
```

```

private fun updateHitTest(): Boolean {
    val frame = arFragment.arSceneView.arFrame
    val point = getScreenCenter()
    val hits: List<HitResult>
    val wasHitting = isHitting
    isHitting = false
    if (frame != null) {
        hits = frame.hitTest(point.x.toFloat(), point.y.toFloat())
        for (hit in hits) {
            val trackable = hit.trackable
            if (trackable is Plane
                && trackable.isPoseInPolygon(hit.hitPose)) {
                isHitting = true
                break
            }
        }
    }
    return wasHitting != isHitting
}

```

```

private fun updateTracking(): Boolean {
    val frame = arFragment.arSceneView.arFrame
    val wasTracking = isTracking
    isTracking = frame.camera.trackingState ==
        TrackingState.TRACKING
    return isTracking != wasTracking
}

```

```

private fun getScreenCenter(): Point {
    val view = findViewById<View>(android.R.id.content)
    return Point(view.width / 2, view.height / 2)
}

```

```

private fun addObject(model: Uri) {
    val frame = arFragment.arSceneView.arFrame
    val point = getScreenCenter()
    if (frame != null) {
        val hits = frame.hitTest(point.x.toFloat(), point.y.toFloat())
        for (hit in hits) {
            val trackable = hit.trackable
            if (trackable is Plane
                && trackable.isPoseInPolygon(hit.hitPose)) {
                placeObject(arFragment, hit.createAnchor(), model)
                break
            }
        }
    }
}

```

```

        private fun placeObject(fragment: ArFragment, anchor: Anchor,
model: Uri) {
ModelRenderable.builder()
        .setSource(fragment.context, model)
        .build()
        .thenAccept {
addNodeToScene(fragment, anchor, it)
        }
        .exceptionally {
Toast.makeText(this@MainActivity, "Error",
Toast.LENGTH_SHORT).show()
return@exceptionally null
        }
    }
}

```

```

        private fun addNodeToScene(fragment: ArFragment, anchor:
Anchor, renderable: ModelRenderable) {
val anchorNode = AnchorNode(anchor)
        // TransformableNode means the user to move, scale and rotate
the model
val transformableNode =
TransformableNode(fragment.transformationSystem)
transformableNode.renderable = renderable
transformableNode.setParent(anchorNode)
fragment.arSceneView.scene.addChild(anchorNode)
transformableNode.select()
    }
}

```

Қорытынды

Осылайша «Медициналық құрал-жабдықтарды жинау және жөндеу үшін толықтырылған нақтылық қосымшасын дайындау» медициналық құрылғылардың 3D моделін жасап осы бағдарламаға енгізу арқылы кодта сілтеме береміз. Бағдарлама медициналық құрылғыларға қызмет көрсету жұмысшыларына арналады. Осы бағдарлама арқылы алыс қашықтықтағы мамандар тікелей мекеме ішіндегі жұмысшыларға кеңес бере алады. Себебі олар бағдарлама ішінде сәйкес медициналық құрылғыны бір уақытта қарастырып ақылдаса алады. Сонымен қатар жұмысшылар медициналық құрылғының бөлшектерімен планшет немесе ұялы телефон арқылы қарап жөндей алады.

Әдебиеттер

1. Алекс Лайтман, Бретт Кинг, Дж. Эпоха дополненной реальности. - М. Изд-«Олимп-Бизнес». Москва, 2018. – 334 б.
2. [Электрондық ресурс], URL: <https://www.plugandplaytechcenter.com/resources/augmented-reality-healthcare/>
3. [Электрондық ресурс], URL: <https://archer-soft.com/en/blog/how-augmented-reality-used-medicine>
4. [Электрондық ресурс], URL: [http://www.tadviser.ru/index.php/Статья:Дополненная_реальность_\(AR,_Augmented_Reality\)#](http://www.tadviser.ru/index.php/Статья:Дополненная_реальность_(AR,_Augmented_Reality)#)
5. [Электрондық ресурс], URL: <https://thinkmobiles.com/blog/augmented-reality-medicine/>
6. [Электрондық ресурс], URL: <https://nanobotmedical.com/augmented-reality/>